

S/N TCφAφ932

LDOS 5.3 Update

The LDOS 5.3 release is a major enhancement from the 5.1.x release of LDOS. Although there have been improvements to most library commands, utilities, and BASIC, it is not prudent at this time to provide documentation as replacement pages to your user manual. You are being provided documentation covering the changes since the 5.1.4 release. A completely new user manual may be made available at a future time. Documentation of a technical nature used by programmers has appeared in our fall 1986 issue of THE MISOSYS QUARTERLY, available on a subscription basis. There may be a file named "README/TXT" on the LDOS disk. If so, this file will contain important information which may not appear in this printed documentation. You should read this file by issuing the command:

LIST README

LDOS 5.3 Installation instructions

The LDOS 5.3 release is an upgrade of the LDOS 5.1.4 operating system. Several important changes have been made to extend and enhance the operating system and its utilities. The date ranging has been expanded to accept dates from January 1, 1980 through December 31, 1999. The directory information for a file will now include a modification time stamp (hours and minutes) in addition to the date stamp. The DATECONV/CMD program is provided to update versions 5.1.4 or earlier LDOS disks to the 5.3 style of dating. Note that this directory convention is that used in LS-DOS release 6.3. The "access" password has been eliminated from the system in order to allow ample directory space for the date and time extensions. The "update" password has been retained. In order to convey a naming convention similar to LS-DOS 6.x, this password is now termed the "owner" password; thus, any reference to the "update" password is assumed to reference the "owner" password and vice versa.

LDOS 5.3 serial/registration number

Your master disk contains an imbedded serial number which is displayed (between the disk name and disk date) when booting the disk. This serial number will be maintained on every BACKUP copy of your LDOS disk. It is this serial number which should be entered onto the accompanying registration card. You should also write down the number in your manual for reference in case you have to contact MISOSYS for support. Please be aware that the DOS performs an integrity check of this serial number; thus, it must not be tampered with.

Additions/changes to included files

The following is a brief description of files added since 5.1.4:

BASIC/CMD	- The disk BASIC interpreter previously named LBASIC/CMD
BASIC/HLP	- A file of HELP information for BASIC
BASIC/OV4	- A BASIC overlay to dump a list of active variables
DATECONV/CMD	- A utility to update LDOS disks earlier than 5.3
DOS/HLP	- A file of HELP information for DOS
HELP/CMD	- A utility used to access the ???/HLP files
TED/CMD	- A full screen ASCII text editor

Backing up your master LDOS disk

It is recommended that you make several backup copies of the 5.3 master disk. The simplest way in a two-drive system is to boot up the 5.3 disk, insert a blank (or otherwise reusable) disk in drive 1, and type the command:

```
QFB :0 :1
```

This procedure assumes that both drives 0 and 1 are 40 cylinder drives. When the copy finishes, you can insert another destination disk and duplicate another copy.

IMPORTANT NOTE

It is important that you do NOT switch between versions 5.3 and any earlier version system disk in drive 0 without rebooting your computer. Also, once you have updated a disk with the DATECONV/CMD program, do not write to that disk using an earlier release of LDOS. As noted above, LDOS 5.3 data disks are completely compatible with LS-DOS 6.3 data disks. Either may be read from or written to with either LDOS 5.3 or LS-DOS 6.3.

Updating your existing disks

To update your existing LDOS 5.1.x disks, boot up with a fresh copy of your new version 5.3 LDOS disk. The LDOS 5.3 disk should remain in drive 0 throughout the update procedure. There are two separate methods for updating 5.1 floppy disks, depending on whether they are system disks or data disks.

Updating DATA disks

Place the DATA disk in drive 1 and issue the command:

```
DATECONV :1
```

This completes the conversion of a DATA disk.

Updating SYSTEM disks

Updating system disks will take two or more steps. Place the 5.1 disk in drive 1, and issue the command:

```
BACKUP :0 :1 (I,S,OLD)
```

If you want to use any of the new files included with the 5.3 release, move them to drive 1 with a command of the form:

```
BACKUP filespec :1
```

where "filespec" designates the new file you wish to copy. Once the BACKUPS are completed, issue the command:

```
DATECONV :1
```

This completes the updating of a SYSTEM disk. Note that Model 4P users must ensure that the file MODELA/III is on your working LDOS system disk. The MODELA/III file is distributed with your TRSDOS 6.x disk.

IMPORTANT NOTE

If you have a sysgened configuration file on your old (now updated) SYSTEM disks, you will have to boot that disk while holding down the <CLEAR> key to inhibit the old configuration; then you will have to recreate your desired configuration once again and save it [using SYSTEM (SYSGEN)].

Updating Radio Shack HARD disks

For most hard disk installations, the following instructions can be used. They assume a 2-head 5 Megabyte hard drive partitioned as 2 drives (0 and 1) and 2 floppy drives (2 and 3). Other installations should be able to adapt these instructions to their specific hardware configuration.

1. Copy your hard disk driver file from your old LDOS disk to a backup copy of your new LDOS 5.3. For a Radio Shack hard drive, this file is named either TRSHD3/DCT or TRSHD5/DCT.

2. Manually invoke the driver with the command:

```
SYSTEM (DRIVE=2,DRIVER="TRSHDx")
```

Respond to any driver queries as you have done in the past. This may include "drive select address" answered as 1; "number of heads for this partition" answered as 1; "starting head" answered as 1.

3. Next, bring up the second head with the command:

```
SYSTEM (DRIVE=3,DRIVER="TRSHDx")
```

As before, this is drive select address 1, 1 head for the partition, but starting head is answered as 2.

4. Now BACKUP all of the files from the floppy SYSTEM disk to the old hard drive's SYSTEM partition (assumed to be head 1 currently assigned to drive 2) with the command:

```
BACKUP :0 :2 (S,I)
```

5. Update the date convention of the hard drive partitions via the commands:

```
DATECONV :2  
DATECONV :3
```

6. Now switch the hard drive SYSTEM partition to drive 0 and the data partition to drive 1 via the commands:

```
SYSTEM (SYSTEM=2)  
SYSTEM (DRIVE=1,SWAP=3)
```

7. Now install the new configuration onto the BOOTing floppy via the command:

```
SYSTEM (SYSGEN,DRIVE=2)
```

Note that you may also install any other sysgenable filters and drivers prior to issuing this SYSTEM command (i.e. KI/DVR, MINIDOS/FLT, etc.).

IMPORTANT NOTE

If you ever have to re-FORMAT a hard drive using the TRSFORMx/CMD utility, you will need to use the DATECONV/CMD utility after the formatting is complete. This ensures that the hard drive partition is marked as using the extended dating facility.

File passwords

Passwords have been applied to most files on your LDOS disk. This protection is for your benefit and will avoid inadvertant deletion or overwrite of a needed file. For your information, the following owner passwords have been applied to all password protected files:

- SYSTEM - All files with an extension of "/SYS"
- BASIC - All files with a filename of "BASIC"
- FILTER - All files with an extension of "/FLT"
- DRIVER - All files with an extension of "/DVR"
- HELP - All files associated with the HELP facility
- UTILITY - All other files with an extension of "/CMD"
- RSOLTOFF - A master password for all files

LDOS customer support

Support for this LDOS product is available from MISOSYS, Inc., PO Box 239, Sterling VA 22170-0239. Our telephone number is 703-450-4181. Hours of operation are 9am through 6pm Monday through Friday Eastern Time. Please have your LDOS registration number available when you call or write for support.

LDOS Version 5.3 update information

This section describes the changes to existing commands, and explains new commands and utility programs. You can continue to use your 5.1.x disks with version 5.3 as long as you update them as explained in the installation instructions. The following is a brief guide to version compatibility.

Disks from earlier versions of LDOS should be converted with the DATECONV utility program before being used with 5.3. Unconverted disks will work but the time and date information will not be correct. Disks formatted and written to by a DOS other than LDOS should first be REPAIred (or use CONV) then processed with DATECONV. If a disk written to with 5.3 is used with an earlier version of LDOS (or some other DOS), it may appear to have an unknown password. If the file was previously password protected, access may not be possible without knowing the owner password. To prevent this, use the COPY library command with the (CLONE=NO) parameter when moving files from version 6.3 to earlier versions or other DOS disks.

You should not use an LDOS 5.3 disk with earlier versions of LDOS nor any version 6 TRSDOS except LS-DOS 6.3 (or later).

LDOS 5.3 disks may be freely used with LS-DOS 6.3 disks.

BOOT-up changes

Since version 5.3 now stores a modification time as well as a date, you will be prompted for the time when booting the computer. You can enter the time in the format, "hh:mm:ss", according to the prompt. If you wish to suppress this time prompt, issue the command:

SYSTEM (TIME=OFF)

when your BOOTing floppy is the SYSTEM drive. The time prompt may be re-activated by the command noted above if "OFF" is replaced with "ON".

LIBRARY command changes

ATTRIB - The UPDATE password field has been redesignated as the OWNER password field - this is a name change only. The access password field has been removed from all files, so the ACC parameter is no longer valid. The DOS now assumes a blank access password for all files. If a protection level has not been assigned to a file, full access will automatically be granted regardless of any owner password. The OWNER password will still be required for full access on password protected files that have a protection level other than FULL. To have a file that allows no access whatsoever without the use of the OWNER password, change the protection level to "NO"

To refresh your mind, here is the command syntax for ATTRIB:

ATTRIB filespec (Owner=a,Prot=b,Vis/Inv)

ATTRIB :d (LOCK,UNLOCK,MPW="aa",NAME="bb",PW="cc")

Parameters for filespec attributes:

Owner= the new owner password

Vis makes file "visible"

Inv makes file "invisible"

Prot= the new protection level:

[NONE, EXec, REad, WRit, NAME, KIll, FULL]

Parameters for disk ATTRIBs:

LOCK Locks the current disk master password onto all visible, non-system files not already protected.

UNLOCK Removes the password from all visible, non-system files with passwords matching the disk master password.

MPW= Specifies the disk's current master password.

NAME= Changes the disk name.

PW= Changes the disk master password.

AUTO - This command has had several improvements. You can now install, display, or invoke an AUTO command on any drive - not just the SYSTEM drive. The syntax is now:

AUTO [=|?][:d][*][dos-command]

=:d executes the AUTO command on :d.

? displays the AUTO command currently on :d.

:d the designated drive otherwise drive 0 is used.

***** Disables the <ENTER> and <BREAK> keys during BOOT.

dos-command any DOS command up to 31 characters in length.

CLOCK - The clock command has been moved to the TIME library command. The new syntax to enable or disable the screen display of the real time clock is:

TIME (CLOCK[=ON|OFF])

CLS - This command has been added to the library. It clears the video screen. Its syntax is:

CLS

COPY - When copying from a 5.1.x or earlier version disk to a 5.3 disk, the old access password, if any, will be removed and the 5.3 style date/time information will be established, the time being set to 00:00:00.

Note: COPY should NOT be used to move System (/SYS) files from one disk to another. Use BACKUP for this purpose.

CREATE - A new parameter, fill, has been added. This parameter allows you to specify a particular character to propagate throughout the created file. To refresh your mind, here is the syntax of this command:

Lrl=l Sets the Logical Record Length (1-256) [256].

Rec=r Allocates r records of LRL length.

Size=s Allocates s K-bytes of space in K (1024 bytes).

SIZE may not be specified if LRL or REC are.

Fill=f The created file will be populated with this byte.

If omitted, the file is not initialized with any byte.

DATE - The acceptable range of dates is from January 1, 1980 through December 31, 1999.

DEBUG - The DEBUG command has been improved to allow for the removal of the extended debugger from resident memory using the command:

DEBUG (OFF,Ext)

provided the memory-resident module was the most recent module installed into memory.

DEVICE - The DEVICE command has had both its display and its operation improved. A new parameter, N, has been added and the display has been paged. To refresh your mind, here is the command's syntax:

DEVICE [(parms)]

B=OFF Suppresses the "byte I/O" portion of the display.

D=OFF Suppresses the drive portion of the display.

N=ON Set non-stop display mode. Assumed if P is selected.

S=OFF Suppresses the options status portion of the display.

P=ON Duplicates device status to the printer.

DIR - The DIRectory command has been considerably enhanced. The display now includes the time stamp and byte EOF position for each file. Also, certain characteristics of the disk itself are displayed. A replica of this display is:

```
Drive d diskname diskdate cyl dh Free=fffff.f/sssss.s Fi=mmm/nnn
Filespec      Attrib  LRL #Recs EOF DE File Size  MOD Date Time
-----
filename/ext SIP*+ pp lrl rrrrr eof de s=sssss.s dd-mon-yy hh:mm

d          drive number
diskname   name of the disk
diskdate   date of the disk
cyl        number of cylinders on the disk
d          disk density: S=single, D=double, H=hard
h          Number of heads on the disk
fffff.f    Amount of free space in K
sssss.s    Total formatted space in K
mmm        Number of free file slots
nnn        Total number of file slots
filename    The name field of a file
ext         The extension field of a file
SIP*+      Attributes indicating: Sys, Inv, Prot, PaDS, Mod
pp         The protection level: NO, EX, RE, WR, NA, KI, FU
lrl        The file's logical record length
rrrrr      The number of records in the file
eof        The position of the last byte in the last sector
de         The number of directory extents occupied by the file
sssss.s    The total space used by the file's records
dd-mon-yy  The date of last modification
hh:mm      The time of last modification, 00:00-23:59
```

DO - The DO command has been improved. First, parameter strings of any ASCII character are acceptable by enclosing them within double quotes. Second, JCL compilation will search for the first available drive which is not write protected for writing the SYSTEM/JCL file.

FORMS - This command has been added to allow you to display or alter the operating parameters of the PR/FLT forms filter once the filter has been installed. The syntax of the FORMS command is:

FORMS [(parms)]

```
Default      Sets all parms to defaults.
Addlf        Adds a linefeed after a carriage return.
Chars=n      The number of characters per printed line.
Ffhard       Use l2d for form feed, rather than linefeeds.
Indent=n     Indents n spaces on lines longer than chars [0].
Lines=n      The number of lines printed per page [66].
Margin=n     Sets the left margin [0].
Page=n       Sets physical page length [66].
Tab          Expands tab characters to tab stops [OFF].
XLATE=X'aabb' Translates aa to bb [00 00].
```

FREE - The display format of the **FREE** command has been improved. It will now display the identical disk configuration information as displayed by the header of the **DIR** command.

KILL - This command has been improved to allow you to specify more than one device specification on the command line. It's new syntax is:

```
KILL filespec[*devspec [filespec[*devspec]
```

LIB - The **LIB** command has been improved to always produce a correct display regardless of the compressed spaces vs special characters state of the video driver.

LIST - This command has been considerably revised. The hexadecimal display mode now produces a combined hexadecimal/ASCII display of a complete 256-byte record. The older format is still available by turning off the "compressed" display mode via the "C=OFF" parameter. Both the hexadecimal and ASCII display modes will produce paged displays, stopping when the display screen fills. Depress <ENTER> for the next page or <c> to continue. The "N" abbreviation for the "NUM" parameter has been eliminated; "N" now allows an override to the paged display to produce a non-stop display. This operation is similar to the "N" parm of the old (and new) **DIR** command. Finally, the "TAB" parameter has been enhanced to accept a tab column number between 1 and 32 with a default of every 8 columns. To refresh your mind, the **LIST** command's syntax is now:

```
LIST filespec[/TXT] [(parms)]
```

Ascii8 Will allow full 8-bit output

NUM Sets line numbering mode for ASCII text.

N=ON Sets non-stop display mode; assumed if "P" is selected.

Hex Sets hexadecimal output format.

C=OFF Turn off compressed display mode for hexadecimal output.

Tab=n Set tab stops and expansion for ASCII list (n=1-32) [8].

P Directs output to the printer.

LINE=n Starts listing at line n in ASCII list.

Rec=n Starts listing at record n in HEX list.

Lrl= Sets logical record length in HEX list.

MEMORY - This command has its display improved to report on the modules resident in protected memory. This requires that all installed modules adhere to the LDOS design specifications of memory resident modules which has been a part of the LDOS system since January 1, 1982. An added "P" parameter allows for printing the memory map.

ROUTE - This command has been improved in two ways. First, a new parameter, **REWIND**, has been added which allows you to start the routed file at its beginning. Second, the command has been improved by allowing it to reuse the previously installed but currently unused high-memory support module of a previous **ROUTE** of the same device. To refresh your mind, the **ROUTE** command's syntax is as follows:

```
ROUTE *devspec1 [TO] [filespec[*devspec2] [(parms)]
```

NIL Specifies that *devspec1 is routed to a bit-bucket.

Rewind Resets the file pointer of filespec so existing data can be reread or overwritten.

SETCOM - This command has been added to the library. It allows you to change and/or display the parameters of the RS232 device driver after the driver has been installed into memory. It's command syntax is as follows:

SETCOM [(parms)]

DEFAULT	Set to system defaults shown below in brackets.
Baud=n	sets the BAUD rate [300].
Word=n	sets word length, 5-8 bits [7].
Stop=n	sets stop bits, 1 or 2 [1].
Parity=	sets PARITY; ON, OFF, "EVEN", "ODD" ["EVEN"]
BREAK=n	sets logical break character to n [OFF].
DTR=	Data Terminal Ready [ON]
RTS=	Request To Send [OFF]
DSR=	Data Set Ready [IGNORE]
CD=	Carrier Detect [IGNORE]
CTS=	Clear To Send [IGNORE]
RI=	Ring Indicator [IGNORE]

SYSTEM - The SYSTEM command, used to select DOS features, has been improved by the inclusion of additional parameters and enhancement of some existing ones. The ALIVE module can now be removed from memory. The DRIVE parameter now pertains to the SYSGEN parameter so that the configuration may be targeted to a specific drive. The Model III (FAST|SLOW) parameter has been adapted to utilize the Model 4 hardware clock speedup while still maintaining an accurate time clock. Model III users should not specify (FAST) without an appropriate hardware speedup. A SMOOTH parameter has been added for faster disk I/O involving floppy drives aligned precisely to 300 rpm. A SWAP parameter has been added that, with DRIVE, allows you to switch the logical drive assignments of any two drives even with a JCL command line. The SYSRES parameter has been improved to have the system overlay use only the amount of high memory normally used by the overlay when it is resident in the system overlay region. The TRACE command has been removed as a separate LIBRARY command and is now a parameter of SYSTEM. The TYPE parameter was altered to just inhibit the type-ahead operation rather than removing the type-ahead task. Here is the syntax of the changed parameters:

SYSTEM (SMOOTH[=ON|OFF])

Alters the floppy disk driver so that the system interrupts are disabled earlier than what would otherwise occur. This has the effect of providing faster I/O with disk drives precisely aligned to 300 rpm where extra sector retries would be necessary. Note that when SMOOTH is turned ON, you will not be able to type ahead during disk I/O nor will you be able to effectively use dump-to-disk ON with LCOMM even at 300 baud.

SYSTEM (SWAP=s,DRIVE=d)

Exchanges drive "s" with drive "d". The command is functional while JCL is in execution even if one of the referenced drives holds the executing JCL file. If one of the designated drives is the SYSTEM drive and JCL is in execution, the other designated drive must contain a SYSTEM.

SYSTEM (SYSGEN[=ON|OFF][,DRIVE=d])

This command creates or deletes a configuration file. "ON" creates a configuration file on drive d. "OFF" removes the configuration file from drive d. "DRIVE=d" specifies the drive for SYSGEN [default is 0].

SYSTEM (SYSTEM=n)

If the designated drive does not contain the DOS and JCL is in execution, the command will abort, otherwise you will be prompted to enter a SYSTEM disk. This command is equivalent to 'SYSTEM (SWAP=n,DRIVE=0)'.

SYSTEM (TRACE[=ON|OFF])

Displays or removes the Z80 Program Counter in the upper* right corner of the video display.

TIME - The TIME command has been enhanced to provide for turning on or off the video screen clock display. The syntax of this command is:

TIME [hh:mm:ss] [(CLOCK[=ON|OFF])]

hh:mm:ss Sets time to hh hours (00-23), mm minutes (00-59), and ss seconds (00-59).

CLOCK Turns on or off the video clock display.

TOF - This command has been added to the library. It will emit a form feed character (12d) to the *PR device. If the printer is currently unavailable, the command does nothing.

Utility command changes

BACKUP - This utility has been changed to allow it to construct a SYSTEM disk from a formatted DATA disk. The VIS parameter has been dropped. The INV parameter has been altered to designate that files invisible to the directory are to be included as well as visible files. The SYS parameter has been altered to designate that system files are to be included in addition to visible files. SYS also reconfigures a DATA disk into a SYSTEM disk by allocating directory entry codes for /SYS files in the Hash Index Table.

FORMAT - This utility has been modified to generate a DATA disk after formatting. DATA disks reserve only two file slots out of the total number of directory slots available. SYSTEM disks, configured by the BACKUP utility, reserve 14 additional directory slots for /SYS files. This facility provides 14 additional file slots for DATA drives over that previously available under earlier releases of LDOS.

LCOMM - This communications program has been enhanced via the addition of parameter control over the codes used for XON and XOFF. Also, LCOMM now makes use of the new @CMNDR command-and-return vector of LDOS. This feature provides you the ability to access LDOS library commands while running LCOMM. The command mode is available via the keystroke sequence, <CLEAR><SHIFT><Ø>. Here is the syntax for invoking LCOMM:

LCOMM *devspec [(parms)]

*devspec is the RS232 device installed via SET, usually *CL.
XlateS=X'fftt' Translates character ff to tt before sending.
XlateR=X'fftt' Translates received character ff to tt.
NULL=OFF Prevents nulls (ØØ's) from being received.

XON=xxx Sets the XON character to xxx [CTRL-Q].
XOFF=xxx Sets the XOFF character to xxx [CTRL-S].

<CLEAR-8> provides a menu of LCOMM commands while on-line.

<CLEAR><SHIFT><Ø> provides the execution of a DOS library command.

PATCH - This utility has been enhanced to support the required finding of byte matching a particular "F-format" patch line prior to installing a "D-format" patch. A new parameter, OPTION, allows you to force or inhibit the required matching (the default is to not require "F-format" patch lines). Coincidentally, a new parameter, REMOVE, has been added to un-install a "D-format" patch that was installed with the appropriate required matching of "F-format" patch lines. The syntax of the PATCH utility is now:

PATCH filespec1[/CMD] [USING] filespec2[/FIX] (parms)

YANK Nulifies the effects of a previous X-type patch.
REMOVE Nulifies the effects of a previous D/F-type patch.
OPTION=ON Match F records before installing D patch [OFF].
X'hhhh'=' Change file starting at address hhhh to data
Drr,pp= Change file starting at record rr offset pp to data.
Frr,pp= Finds data at record rr byte offset pp.
Lnn Designates DOS library command ISAM number.

data hexadecimal values of the form: xx xx xx ...
 ASCII values of the form: "abcdefg..."

TED - This is a new text editor which has been added to LDOS. It is a full-screen ASCII text editor with many editing features. It provides the facility of editing ASCII text files; it is very useful for those LDOS users not already owning a text editor or word processor. Documentation for TED is a separate section of this update documentation.

KI/DVR - The keyboard driver has been improved to allow the key combination, <CLEAR><SHIFT><Ø>, to generate a code of 16Ød - the same as <CLEAR><SPACE>. Note that it will not toggle CAPS LOCK; only <SHIFT><Ø> will toggle CAPS LOCK. Another enhancement to LDOS will restrict KI/DVR from being inadvertently loaded by issuing it as a command; it requires installation via the SET library command.

RS232x/DVR - The serial driver provided with your system has been enhanced to provide for the selection of a received logical BREAK character rather than arbitrarily using a code of Øld. Note that BREAK normally defaults to OFF; thus, no received character will be interpreted as a BREAK for the purposes of the system's keyboard scanner. The driver parameters may also be altered after the driver is installed by using the SETCOM library command documented earlier.

MINIDOS/FLT - This keyboard filter has been improved to allow the "repeat last DOS command" to properly function regardless of the filter's position in the keyboard device chain.

PR/FLT - The PR/FLT filter program has been enhanced to allow its operating parameters to be changed after the filter has been installed into memory. See the FORMS library command previously documented.

HELP - The HELP facility has been added to provide on-line help information for both DOS commands and disk BASIC statements/functions. The HELP displays use the following notations:

[] Optional arguments or parameters
| Alternative selection: ON|OFF means ON or OFF.
< > Key combination you press.

Help screen format:
Brief description
Command Syntax
Parameter descriptions

Help information is retrieved by entering commands of the form:

HELP DOS BACKUP
HELP BASIC INPUT@

DATECONV - This utility program has been added to convert DOS data disks earlier than this 5.3 release to the extended date and time stamp usage. SYSTEM disks must first have the DOS files moved to them via BACKUP then processed by DATECONV.

BASIC enhancements

The disk BASIC interpreter has been improved by several additions and the name has been changed to "BASIC/CMD". First, two more edit functions have been added. These are the ability to copy a BASIC program line to another line number and the ability to move a BASIC program line to another line number. No automatic renumbering of imbedded line number targets is done so you will still have to adjust the targets of any GOTOs, GOSUBs, etc. These two edit additions have been added to the BASIC/OV3 overlay file so that overlay must be present on-line for you to be able to use those two commands. All of the single-key edit commands are reprinted here for you:

<UP ARROW>	Cursor up one line
<DOWN ARROW>	Cursor down one line
<LEFT ARROW>	Cursor to first line
<RIGHT ARROW>	Cursor to last line
<.>	Display current line
<,>	Edit current line
<A> nl[,inc]	AUTO line nl with increment
<C> nl,n2	Copy line nl to line n2
<D> nl[-n2]	Delete line(s) nl through n2
<E> nl	Edit line nl
<L> nl[-n2]	List line(s) nl through n2
<M> nl,n2	Move line nl to be line n2

INPUT@

This provides a much-requested facility of controlling keyboard input to a string variable with prompting and screen field highlighting. Two forms of INPUT@ are supported. One provides the minimal improvement of allowing the input prompt to appear starting at a designated video position - similar to a PRINT@ statement. The second form is more powerful. It allows you to specify an input field width, an input field fill character, whether the input should be alphanumeric or just numeric, and whether the input should automatically terminate when the "field width" number of characters have been entered rather than requiring a hard <ENTER>. The two forms of INPUT@ are:

```
INPUT@pos[,"message"];var[,var]
INPUT@pos[,"message"],fw,"$|#[*][f]";var$
```

pos	The screen position for the message or field.
var	Variables used to store the data retrieved.
fw	The maximum input field width <1-240>.
\$	Specifies acceptance of characters <32d-127d>.
#	Specifies acceptance of characters <0-9, period, minus, plus>.
*	Specifies immediate <ENTER> on maximum input.
f	Designates a field fill character ['_'].
var\$	The string variable to receive the input.

When using the second form, the single input variable will always be a string variable and the input, regardless of designated type, will always be a string of characters. Note that since the "immediate <ENTER> on maximum input" character is examined before the programmed fill character, you cannot designate an asterisk as the fill character unless you specify the forced immediate <ENTER>. Thus, "***" will be interpreted as accepting numeric only, immediate <ENTER> on full field, and use an asterisk as the fill field character. On the other hand, "#*" will default the field fill character to an underline.

CMD"V" - This BASIC extension can be used to dump a list of active variables and their values and user defined functions while a program is running (or after it was interrupted or ended). The syntax of this command is:

```
CMD"V [*DO|*PR] [-S] [-A] [=x]"
```

*DO *PR	Designates the output to either device [*DO]
-S	Restricts output to scalars only.
-A	Restricts output to array variables only
=x	Restricts output to variables starting with "x"

TED - ASCII Text Editor

The Text Editor (TED) is a full screen "quick" text editor with typical word-processing type features (four-directional cursor movement; two-directional scrolling; text insertion and overstrike; string search and replace; block copy, delete, and move; directional delete; large text buffer; etc); however, TED was not designed to be a full featured word processor. TED was designed for you to be able to rapidly enter a full-screen text editing environment while accomplishes many of your text file editing tasks.

Summary of editing commands

The following command summary is presented for your use. These are the command keys and their functions as supplied by TED. Once you become familiar with the operation of TED, this section may be all you need to refer to from time to time to jog your memory. Here are TED's commands.

Key Entry	Function
<^A>	Toggle overstrike/insert modes
<^B>	Specify BLOCK...
<^D>	Specify DELETE
<^F>	FILE the text buffer to disk
<^G>	GO find the next search string match
<^H>	Same as the <LEFT ARROW> key
<^I>	Same as the <RIGHT ARROW> key
<^J>	Same as the <DOWN ARROW> key
<^K>	Same as the <UP ARROW> key
<^L>	LOAD a text file into the buffer
<^M>	Same as the <ENTER> key
<^N>	Go to the NEXT video page
<^R>	REPLACE searched string with new string
<^S>	SEARCH for a string
<^U>	Go UP to the previous video page
<CLEAR SHIFT =>	EXIT the text editor
<LEFT ARROW>	Move the cursor one position left
<RIGHT ARROW>	Move the cursor one position right
<DOWN ARROW>	Move the cursor one position down
<UP ARROW>	Move the cursor one position up
<SHIFT LEFT>	Move the cursor to the beginning of the line
<SHIFT RIGHT>	Move the cursor to the end of the line
<^Z>	Move the cursor to the end of the text
<SHIFT UP>	Move the cursor to the beginning of the text

Invoking TED

TED is invoked via the command:

```
TED [*|filespec[/TXT]]
```

TED will display a welcome message on the 16th line of the video screen. This display line will also be used for the display of status, prompting, and error messages. TED displays three different types of messages during its operation. Error messages are indicated by a terminating exclamation point, "!". Queries which need a response are indicated by a terminating question mark, "?". Informative messages use no special character for their termination. Thus, "Marker!" is an error, "String?" is a query, and "Block" is information.

The first fourteen lines of the video screen are used as a display window for the text file under edit. This text area is separated from the status display line by a horizontal line drawn with the underbar character.

Throughout this documentation, keystrokes are denoted by being enclosed in angle brackets. Most of TED's commands are entered with the control key (<SHIFT><DOWN ARROW> under LDOS). Some use the <CLEAR> key, instead. Throughout this documentation, the control key will be noted by the caret character, "^". Thus, for example, the "block" command will be denoted as <^B>.

You can re-capture the contents of the text buffer after you have terminated TED by putting an asterisk on the command line. Similarly, if a filespec is entered on the command line, it will be loaded into the text buffer automatically just as if you entered a subsequent LOAD command.

Text entry modes

TED will accept only displayable ASCII characters in the range 20H through 7FH for text entry. Any other character value will be interpreted as a command entry. If it matches a value in the command table, that command will be invoked; otherwise, the entry will be ignored.

TED operates in two text entry modes: overstrike and insert. The initial mode established when TED is first invoked is the "overstrike" mode. TED can be modified to initially enter the "insert" mode, if you so desire, by changing the value of a single byte in the application file itself. The method of accomplishing this is explained in the section covering alterations to TED.

While TED is in "overstrike" mode, it will use an underbar as the cursor character (character value 5FH). When you toggle to "insert" mode, the cursor is changed to a full graphics block (character value BFH). You toggle from one mode to the other via the <^A> command.

When TED is in "overstrike" mode, any acceptable text entry typed character is written over the character which appears under the blinking cursor. You can overstrike a newline character (i.e. <ENTER>, which is displayed as character value 84H). You can also overstrike either a "begin" block marker (character value B7H) or an "end" block marker (character value BBH). You can be in overstrike mode when you come to the end of the text (or starting from an empty text buffer, for that matter) and still be able to enter text in this mode.

When you switch to "insert" mode, anytime you enter an acceptable text entry character, the entire text will be pushed down one position starting from the character under the cursor to make room for the inserted character. The video screen will be constantly updated as text is inserted.

The text entry mode is only changed via the <^A> command. Going into "delete" mode does not change the mode of text entry.

Internally, TED uses a NULL character (character value 00H) to indicate the terminating position of the text buffer. Throughout this documentation, the word "NULL" denotes this facility

Exiting from TED

It is easy to exit TED and return to DOS Ready. The <CLEAR SHIFT => command tells TED you wish to exit. If the text buffer is empty, TED will immediately terminate. However, if there is any text in the buffer, you are provided an opportunity to retract your request. TED will display the prompt message,

Press <ENTER> to EXIT

If you wish to exit TED, all you need do is depress the <ENTER> key and TED will terminate. Any other keystroke entered in response to the prompt will be interpreted as a retraction of the EXIT request.

If you inadvertantly exit TED and wish you hadn't because you forgot to save the text, you may be in luck. Just issue the command: TED *.

Loading a text file

The <^L> command is used to load a text file into the text buffer area. When you depress <^L>, you will be prompted for the name of the file via the prompt message,

Filespec?

If the file you wish to load has an extension of "/TXT", you do not have to enter the extension. If the extension is omitted from your entry, "/TXT" will be automatically provided.

The LOAD command will not automatically clear any text remaining in the text buffer prior to the LOAD. This means that the new text from the disk file will be concatenated to the old text in the editing buffer. The new text is not inserted at the cursor position but rather is appended to the end of the current text. If you wish to load the new file over the old text, simply invoke the command sequence, <SHIFT UP ARROW> followed by <^D> then <SHIFT DOWN ARROW Z>. This will delete the entire text buffer. Alternatively, you can invoke a <SHIFT DOWN ARROW Z> then follow it with a <^D> then <SHIFT UP ARROW>. Both sequences will delete all of the text in the buffer. You can then load in the new file.

If the file is too large to fit into the available text buffer, the error message,

No room!

will be displayed and no text will be loaded. If any disk read error is encountered while reading the text file into the text buffer, the DOS error message will be displayed in the status line. The text which was loaded up to the point of encountering the error will be retained in the text buffer.

Entering text

Entering text is easy, you just type away. If you already have text in the buffer and wish to enter new text at the end, just move the cursor to the bottom (via the <SHIFT DOWN ARROW Z> key), then type in your text. If you wish to enter new text at some other point, just position the cursor, toggle to the "insert" mode, then type away. TED will stay in "insert" mode until explicitly toggled back to "overstrike" mode.

As you are entering text, any word which is too long to fit at the end of a video line will be split at the 80th column and continued onto the next line. These "long words" are not automatically bounced onto the subsequent line, as is the case with the typical "word processor".

TED also provides you with other facilities for editing text. You can perform a "cut and paste" operation by first marking a block of text and then moving it to where you want the marked block positioned. You can "search and replace" a find text string of up to 23 characters with a replacement text string of up to 23 characters. Sorry, but due to space restrictions, the search is case sensitive.

Cursor positioning manipulations

In all cases concerning cursor manipulation, the last position of a line is interpreted according to the following priority:

1. the NULL which is used by TED to indicate the end of the text;
2. a new line character, the <ENTER>;
3. or the 64th character on the line.

This means that when TED needs to determine what is the last displayed character position on a line, it will first check for case 1. Failing that, it will check for case 2. Failing that, the last character position will be the 64th column, by default. TED provides you with many commands to position the cursor and/or display different portions of your text. The ARROW keys are the primary tools to move the cursor. These keys will be interpreted as cursor movement requests unless TED is in the DELETE or BLOCK modes.

The <LEFT ARROW> moves the cursor one position to the left for each depression (or repeated key, if held down). When the cursor is positioned over the first character of a line, a <LEFT ARROW> request will move the cursor to the last character position of the previous line.

The <RIGHT ARROW> request will move the cursor one position to the right. When the cursor is positioned over either a new line character or the 64th character position on the line, a <RIGHT ARROW> request will move the cursor to the first column of the succeeding line. TED will not advance the cursor past the terminating NULL.

The <DOWN ARROW> request will move the cursor to the succeeding line at the same column position. However, if that position would be beyond the last position of that line, the cursor will be repositioned to the last position of that line. If the line position of the cursor had been the last line of the text portion displayed on the video screen, the text will be scrolled up one row to make the succeeding line visible. TED will not advance the cursor past the terminating NULL.

The <UP ARROW> request will move the cursor to the preceeding line at the same column position. However, if that position would be beyond the last position of that line, the cursor will be repositioned to the last position of that line. If the line position of the cursor had been the first line of the text portion displayed on the video screen, the text will be scrolled down one row to make the preceeding line visible. Obviously, if the first line displayed was the first line of your text, the <UP ARROW> request to change the cursor position will be ignored.

The <SHIFT LEFT ARROW> request will move the cursor to the first position of the current line. The <SHIFT RIGHT ARROW> request will move the cursor to the last position of the current line. You can position the cursor to the first position of the text buffer by a <SHIFT UP ARROW> request. If that text position had not been on the video display, the screen will be refreshed so that the top of the text will be displayed starting from the top of the video display. Finally, the <SHIFT DOWN ARROW Z> will refresh the screen so that the line which contains the terminating NULL will be displayed at the top of the video screen and the cursor will be positioned over the NULL.

The page up, <^U>, command will refresh the video screen so that the new first displayed line is twenty one lines previous to the current first displayed line. That also means the new last displayed line was the first line displayed prior to the page up request. This all assumes that there are at least twenty one lines preceeding the top line. If there are fewer than twenty one lines, the result will be as if you had invoked a TOP command, <SHIFT UP ARROW>.

The page next, <^N>, request will refresh the video screen so that the new first line displayed is the last line of the current displayed text. If the video display has fewer than 14 lines of text displayed, the page next request will be ignored.

After either a page up or page next request, the new cursor position will be the home position of the screen (i.e. row 0, column 0).

Text deletion

TED provides five forms of text deletion in addition to the block deletion discussed later. To delete the single character which appears under the cursor, invoke the delete command via <^D>. This action will get rid of the character and all text which succeeded that character will be pulled back one position. The <^D> command also puts you into DELETE mode which is made apparent by the display of the word,

Delete

in the status line. The DELETE mode is active for only the next keyboard entry. There are only four subcommands associated with the DELETE mode: delete to beginning of line (bol), delete to end of line (eol), delete to top, and delete to bottom. These subcommands are specified by the cursor movement keys associated with cursor positioning. To refresh your memory, use the following table:

Deletion desired	Command sequence
delete to bol	<^D> then <SHIFT LEFT ARROW>
delete to eol	<^D> then <SHIFT RIGHT ARROW>
delete to top	<^D> then <SHIFT UP ARROW>
delete to bottom	<^D> then <SHIFT DOWN ARROW Z>

After the <^D> is requested, the character now under the cursor is the character which was to the right of the deleted character. Since in the case of delete to bol and delete to top, you are deleting text which is in front of the cursor, you really don't want to delete the character which is under the cursor after the <^D>. Well, you don't have to worry about that because those two subcommands properly backup one position before continuing the deletion.

By the way, TED considers all four of these subcommand deletions as severe; thus, it will issue the "Press <ENTER> to confirm" query and expect the entry of <ENTER> in order to carry out the deletion.

Block operations

The BLOCK command, <^B>, has five subcommands: Begin, End, Copy, Delete, and Move. These subcommands are specified by entering the first letter of the subcommand word (, <E>, <C>, <D>, or <M>). The entry may be in either upper or lower case. Note that these subcommands are NOT control key combinations but normal alphabetic single-key entries. When you invoke the BLOCK command, the word,

Block

will be displayed in the status line.

A block deletion request is considered by TED to be severe enough to warrant double checking your request before going ahead and performing the deletion. Thus, the operation of deleting a block was designed to be a BLOCK-Delete function sequence (<^B>, <D>) rather than a DELETE-BLOCK function sequence (<^D>, <^B>) to avoid the single character deletion which always occurs with the DELETE command.

Anytime you need to deal with a block; say to copy it, move it, or delete it, you have to first mark it. The beginning and ending positions of a block are marked by first positioning the cursor over the first character of the block and then entering the two command sequence, <^B> followed by . This is followed up by positioning the cursor over the character immediately following the last character of the block and then entering the two command sequence, <^B> followed by <E>. The beginning position will be indicated on the display by a "begin" marker which is inserted by TED into the text. The marker is displayed as a graphic left bracket (character value B7H). The ending position will be indicated on the display by an "end" marker which is also inserted by TED into the text. The marker is displayed as a graphic right bracket (character value BBH). These markers occupy ordinary text positions; thus they may be deleted or overstricken. Any remaining in the text buffer at the time a FILE command is performed will be written to the disk file just as if they were ordinary text characters [internally, TED uses the value FEH for a "begin" marker and the value FFH for an "end" marker].

Although you can mark as many blocks as your heart desires, TED provides no way to differentiate between marked blocks in other than the BLOCK-DELETE function. For copying and moving blocks, the first block marked in the text is the one chosen for copying or moving. On the other hand, a BLOCK-DELETE request requires that the cursor be positioned within the interior of the marked block which is to be deleted.

To COPY the first marked block in the text to some other position, simply mark the beginning and end of the block as discussed above, move the cursor to the position in the text where you want the marked block copied into, then invoke the block copy command via the sequence, <^B> followed by <C>. Note that the block which will be copied is the first marked block found in the text buffer. A few things could go wrong with your request. If TED can find no properly marked block, it will display the error message,

Marker!

and terminate the block mode. Another error which could occur is when the position you wish the block copied into happens to be in the interior of the block itself! Such a block copy can not occur. You will be informed of this by a display of the error message,

Cursor!

The successful block copy operation only copies the marked text; the markers are not copied as well. In fact, the marked text remains in its original position relative to the text which surrounds it. The cursor position relative to the text will be unchanged after the block is copied; however, the screen may be refreshed and the physical location of the cursor on the screen may be different.

A block of text may be MOVED from one position to another by a command sequence similar to the block copy. In this case, simply mark the beginning and end of the block as discussed above, move the cursor to the position in the text where you want the marked block moved to, then invoke the block MOVE command via the sequence, `<^B>` followed by `<M>`. Again note that the block which will be moved is the first marked block found in the text buffer. This operation is essentially one of copying and automatic deleting without the double check prompt. As in the case of the block copy, the same errors are possible with similar diagnostic messages when things are not as they should be. With the block move command, the new cursor position will be the new position of the moved block. The screen may be refreshed and the physical cursor position altered to accomodate this request.

The last block operation is deletion. Similar to the above functions, you first must mark the block's beginning and ending positions. You must then position the cursor to the interior of the marked block and invoke the command with the sequence, `<^B>` followed by `<D>`. If TED is confident that the cursor position is interior to a marked block, it will double check your request by issuing the prompt,

Press `<ENTER>` to confirm

It is necessary to depress `<ENTER>` to affirm your intentions. Any other character entry (including a "Y") will cause TED to ignore the block delete request.

The same errors as for copy and move can occur; however, the messages may not be for the same reasons. When a block delete is requested, TED will first look for an ending block marker starting from the cursor position. If none is found, the error displayed will be "Marker!". This doesn't mean necessarily that a properly marked block is missing. On the other hand, if an ending marker is found past the cursor position, TED next scans forward for a beginning block marker. A "Marker!" error will also be posted if none is found. If a marker is found but is also past the cursor position, a "Cursor!" error will be posted. The remaining situation is the correct one; the cursor is positioned interior to the marked block and that block will be deleted. The screen will be refreshed and the cursor will be moved to the relative position of the block which was deleted.

Filing away your text to a disk file

The `<^F>` command is used to FILE the contents of the text buffer area into a disk file. When you depress `<^F>`, you will be prompted for the name of the file via the query message,

Filespec?

If the file specification you wish to use has an extension of `"/TXT"`, you do not have to enter the extension. If the extension is omitted from your entry, `"/TXT"` will be automatically provided.

The FILE command will save the entire text buffer, excluding the terminating NULL but including any block markers, into the disk file identified by your input. If any disk write error is encountered while saving the text buffer into the disk file, the standard DOS error message will be displayed. In any case, the text buffer is left undisturbed.

Text search

TED provides the SEARCH command to scan the text buffer for a specified string of characters. You specify the search by invoking the command with <^S>. TED then prompts you for the search string with the query message,

String?

You can enter up to 23 characters to be used for the search string. Terminate your search string with an <ENTER> (the <ENTER> character code is not included as one of the 23 characters). TED will then look for the string starting with the first character immediately following the cursor. The matching is case sensitive which means that characters entered in upper case must be found in upper case and characters entered in lower case must be found in lower case. If the search string cannot be found, the message,

Can't!

will be displayed. At this point, the cursor location remains unchanged. If, on the other hand, a matching string of text is found in the text buffer, it will be displayed. The display window will be redrawn starting with the line which contains that string. The cursor will be repositioned to the first character of the matching string.

If you enter a null string in response to the "String?" query, then the search will proceed with the last entered non-null search string, providing one was available. A null string is, of course, entered by responding to the "String?" query with an immediate <ENTER> keystroke. Using this procedure, you can advance the cursor to each occurrence of the search string in question.

Another way to find each occurrence of a search string is with the GO command, <^G>. Each depression of <^G> is identical to the sequence, <^S> followed by <ENTER>.

Text search and replace

TED also provides the capability of replacing a text string matching up with the search string with a different string - the replacement string. When the REPLACE command is invoked via <^R>, the query message,

String?

will be displayed. Although the message is the same as for SEARCH, this query is asking you for the replacement character string. You can enter up to 23 characters to be used for the replacement string. Terminate your string with an <ENTER> (the <ENTER> character code is not included as one of the 23 characters). TED will then look for the currently pending SEARCH string starting with the first character IMMEDIATELY under the cursor. If the SEARCH string cannot be found, the message,

Can't!

will be displayed. At this point, the cursor location remains unchanged. If, on the other hand, a matching string of text is found in the text buffer, it will be replaced with the REPLACE string. The display window will be redrawn starting with the line which contained the string which was replaced. The cursor will be repositioned to the first character immediately following the replacement string.

If you wish to replace the next occurrence of text which matches up with the SEARCH string with that same REPLACEMENT string, all you need do is invoke the REPLACE command, <^R> followed by a NULL replacement string. A null string is, of course, entered by responding to the "String?" query with an immediate <ENTER> keystroke. Since the cursor is repositioned after a replacement to the first character immediately following the replacement string, your subsequent replacements will not get you into trouble if the SEARCH string happened to be a substring of the REPLACEMENT string!

The GO command, <^G>, still functions to find the next occurrence of the SEARCH string. Knowing this, if the next occurrence of the search string is beyond the text currently displayed on the screen and you wish to confirm its replacement, simply GO to the next occurrence then REPLACE, as necessary.

Printing text

TED provides no facility for printing your text file. The most reasonable way of accomodating that function is with the LIST command provided as part of LDOS.

Text recovery

There may be times when you exit the TED application inadvertantly without filing the edited text to a disk file. Some full-featured word processors permit you to re-enter their editor with an asterisk parameter which generally denotes "reclaim the text buffer". TED also provides this feature. Instead of automatically clearing the text buffer as TED does, "TED *" will display whatever is in the text buffer bmemory area. Thus, if you have not altered any of the information in that memory area, you can always go back and recapture it.

One cautionary note. Since all of the text pointers normally established by TED will not be initialized when invoking TED via the "*", it will be necessary to scroll through the text until reaching its last character prior to doing any other operation. This may also be performed using NEXT PAGE.

Applications	Catalog #	04/01/87	07/01/87
=====			
DSM51	#L-35-204	\$59.95	\$59.95
LED	#L-30-020	\$29.95	\$29.95
QuizMaster	#L-51-500	\$24.95	\$24.95
DSM4	#L-35-205	\$59.95	\$59.95
Little Brother-M4	#L-50-510	\$74.95 B	\$74.95 F **
LBMU-M4	#L-50-515	\$29.95	\$29.95
LS-Host/Term	#L-35-281	\$59.95	\$59.95
LS-LED	#L-30-021	\$29.95	\$34.95 **
Mister ED	#M-51-028	\$59.95 B	\$59.95 B
PRO-WAM	#M-51-025	\$59.95 B	\$74.95 F **
Utilities	Catalog #	04/01/87	07/01/87
=====			
diskDISK	#L-35-211	\$39.95	\$39.95
FASTBACK	#L-30-055	\$29.95	\$29.95
Mark III Collection	#M-99-003	\$99.95 H	\$99.95 H
UTILITY-I	#L-32-070	\$29.95	\$29.95
LS-diskDISK	#L-35-212	\$39.95	\$39.95
LS-FASTBACK	#L-30-056	\$29.95	\$29.95
LS-UTILITY	#L-32-150	\$29.95	\$29.95
Mark IV Collection	#M-99-003	\$99.95 H	\$99.95 H
CP/M products	Catalog #	04/01/87	07/01/87
=====			
UNREL-CPM	#M-32-054	\$39.95	\$49.95 **
LCOPY	#M-32-063	\$39.95	\$39.95
Publications	Catalog #	04/01/87	07/01/87
=====			
LDOS 5.1.4 Manual	#L-40-020	\$30.00 D	\$30.00 D
LDOS QUARTERLY Volume II	#L-49-200	\$9.95 C	discontinued
smallDOS 5.1.4 Manual ...	#L-40-040	\$10.00 A	\$10.00 A
THE SOURCE 3-Volume Set .	#L-60-020	\$99.95 F	\$74.95 F **
NOTES FROM MISOSYS (set)	#M-49-000	\$4.00 A	discontinued
THE MISOSYS QUARTERLY	subscription	\$25 US, \$30 Can, etc.	

Freight codes: A = \$2.50; B = \$3.00; C = \$3.50; D = \$4.00
E = \$4.50; F = \$5.00; G = \$7.50; H = \$10.00
All unmarked are \$2.00 each
Canada/Mexico add \$1 per order
Foreign use US rates times 3 for air shipment

We accept MasterCard, VISA, and Choice credit cards
Checks must be drawn on a US bank.
COD's are cash, money order, or certified check to first-time customers

MISOSYS, Inc., PRICE LIST effective April 1, 1987 and July 1, 1987

** indicates a price change effective 7/1/87

MS-DOS Products	Catalog #	04/01/87	07/01/87	
=====	=====	=====	=====	
DED86	#M-86-020	\$59.95 F	\$59.95 F	(see note)
DSM-86	#M-86-204	\$59.95 F	\$59.95 F	
ED/ASM-86	#M-86-030	\$124.95 F	\$124.95 F	(see note)
FED86	#L-86-010	\$29.95	\$29.95	
FM86	#L-86-050	\$29.95	\$29.95	
Little Brother-MS	#L-86-510	\$74.95 F	\$74.95 F	
LBMU-MS	#L-86-515	\$29.95	\$29.95	
RATFOR-86	#M-86-073	\$99.95 F	\$99.95 F	

Note: DED86 is \$49.95 until 04/30/87

ED/ASM-86 is \$89.95 until 04/30/87

TRS-80 Operating Systems	Catalog #	04/01/87	07/01/87	
=====	=====	=====	=====	
LDOS-I 5.1.4 system	#L-10-010	\$49.95 F	\$59.95 F	**
LDOS-I 5.1.4 disk	#L-10-110	\$14.95	\$24.95	**
LDOS-III 5.1.4 system ..	#L-10-030	\$49.95 F	\$59.95 F	**
LDOS-III 5.1.4 disk	#L-10-130	\$14.95	\$24.95	**
LDOS 5.3 Mod 4 Int. Kit .	#M-12-110	\$29.95	\$29.95	
LDOS 5.3 Mod3 Upgrade Kit	#M-10-033	\$24.95	\$34.95	**
LDOS 5.3 Max Upgrade Kit	#M-10-833	\$24.95	\$34.95	**
LS-DOS 6.2 Model II/12 ..	#M-11-002	\$49.95 B	\$49.95 B	
SOLE	#M-10-021	\$14.95	\$14.95	
FIX Disk II	#L-70-400	\$14.95	\$14.95	
RSHARD - R/S HD driver ..	#M-12-013	\$29.95	\$29.95	

Language Facilities	Catalog #	04/01/87	07/01/87	
=====	=====	=====	=====	
CON80Z	#M-30-033	\$24.95	\$29.95	**
DSMBLR	#M-30-053	\$29.95	\$34.95	**
EDAS	#M-20-082	\$74.95 B	\$74.95 D	**
EnhComp	#M-20-072	\$124.95 B	\$124.95 D	**
HartFORTH	#M-20-071	\$74.95 A	discontinued	
MC	#M-20-064	\$124.95 D	\$124.95 D	
MRAS	#M-20-083	\$74.95 B	\$89.95 D	**
TBA	#L-21-010	\$24.95 B	\$39.95 D	**
LS-TBA	#L-21-011	\$24.95 B	\$39.95 D	**
PRO-CON80Z	#M-31-033	\$24.95	\$29.95	**
PRO-CREATE	#M-21-082	\$74.95 B	\$74.95 D	**
PRO-DUCE	#M-31-053	\$29.95	\$34.95	**
PRO-EnhComp	#M-21-072	\$124.95 B	\$124.95 D	**
PRO-HartFORTH	#M-21-071	\$74.95 A	discontinued	
PRO-MC	#M-21-064	\$124.95 D	\$124.95 D	
PRO-MRAS	#M-21-083	\$74.95 B	\$89.95 D	**
RATFOR-M4	#M-21-073	\$99.95 F	\$99.95 F	
UNREL-T80	#M-30-054	\$39.95	\$49.95	**